

Hot Codes in Photo Mechanic

Webinar Notes

Contents:

[Introduction and Links](#)

[Caveats](#)

[Code Replacement](#)

[What are Hot Codes?](#)

[History Break](#)

[Hot Codes: Code Replacement Evolved](#)

[General Advice](#)

[The Hot Code Template](#)

[Demo #1: Using Hot Codes to route files to specific folders](#)

[Demo #2: Applying metadata to files on ingest based on in-camera Folders](#)

[Demo #3: Applying Metadata to Photos from a Spreadsheet](#)

Introduction and Links

This is not an intro to Photo Mechanic. We'll be assuming you know the basics of looking at images in PM, ingesting from memory cards, using Variables, and editing basic metadata. If you do not know these topics, we suggest you join one of our Getting Started webinars at this link:

<https://home.camerabits.com/getting-started-with-photo-mechanic-webinars/>

Or, view tutorials on these topics at:

<https://docs.camerabits.com>

One particular area you might want to brush up on is the topic of Variables in Photo Mechanic:

<https://docs.camerabits.com/support/solutions/articles/48000207639-introduction-to-photo-mechanic-variables>

The sample Hot Codes template we'll be using in this demo is available as a Google Sheets file at:

https://docs.google.com/spreadsheets/d/17I0-VBZI-Qvdea8UxhyXozOEdejiZMmxv_YdIYdjQWU/edit?usp=sharing

Link to rewatch this webinar after it is over:

https://youtube.com/live/v_y7agK6X_8

Caveats

Hot Codes can seem very complicated. They can be difficult to explain because they involve different series of dynamic, changing information that, in turn, reference other dynamic, changing sets of information. On top of that, different people have different "mental models" of how sets of information are represented in their minds. This can make explanations difficult. You can rewatch this webinar, pause, go back, and change the playback speed if that helps you. But, talking about it can make it seem much more complicated than it actually is in practice, so stay patient.

Code Replacement

To understand Hot Codes, first, we'll review Code Replacements. This is where you can specify short *codes* to be *replaced* by longer text strings. This is useful in many different scenarios, of course, but it is probably most famously used in sports photojournalism. Photographers and editors will download or create Code Replacement files which are plain text files with data separated by tabs.

Note: It is important that Code Replacement files are in plain text, with no hidden formatting characters. But, because spreadsheet programs like Microsoft Excel and Google Sheets are great for working with data in columns, they can be very useful for creating Code Replacement files. When you do this, you simply need to export the spreadsheet as a Tab-Separated Value (.tsv) file. Photo Mechanic can read .txt or .tsv files that are plain text and tab-separated.

The first column will contain the codes, for example, an abbreviation of a team name combined with a jersey or bib number. The next column will contain the name of the player, and subsequent columns can contain more information, like the player's position. Here is an example:

pt1	Jeff Attinella	Attinella	GK
pt10	Sebastián Blanco	Blanco	MF / FW
pt12	Steve Clark	Clark	GK
pt14	Andrés Flores	Flores	CDM
pt16	Zarek Valentin	Valentin	LB / RB
pt17	Jeremy Ebobisse	Ebobisse	ST / W
pt18	Julio Cascante	Cascante	CB
pt19	Tomás Conechny	Conechny	MF
pt2	Jorge Moreira	Moreira	RB

When you export this sheet to a plain-text .tsv, you load it into Photo Mechanic by going to [Edit> Settings > Set Code Replacements](#) and selecting the file. Once the file is loaded into Photo Mechanic, you can make use of the Code Replacements by typing = then the code, then = again. The = is called the delimiter and if you want to change that, you can do so in the Set Code Replacements dialog. So, for example, after you load in the Code Replacement file above, when you type the `=pt12=` into any field in Photo Mechanic, it will be replaced with "Steve Clark." To replace the code with data from other columns, simply add the column number to your code. Examples:

`=pt18=` will resolve to Julio Cascante

`=pt18#2=` will resolve to Cascante

`=pt18#3=` will resolve to CB

For years this feature has helped editors add names to captions, titles, and keywords quickly, without spelling mistakes.

For more documentation on Code Replacement feature in Photo Mechanic, as well as other ways it can be used or combined with Variables in Photo Mechanic go to this link:

<https://docs.camerabits.com/support/solutions/articles/48000223660-introduction-to-code-replacements-in-photo-mechanic>

Pro Tip: When creating Metadata Templates with Code Replacements that you'll save to apply at a later time, you should create them in Photo Mechanic *before* loading the associated Code Replacement files. This way you'll avoid having the Code Replacement evaluated as you're typing it.

What are Hot Codes?

Hot Codes are an advanced type of Code Replacement, where the *code* that gets replaced is not simply based on a static bit of information, but dynamically references a variable in Photo Mechanic. This means that for a *code* that you insert into a field in Photo Mechanic, the *replacement* that gets inserted can ***change based on metadata that is specific to that individual file.***

For example, one “hot” code like [player] inserted into the caption field of one file may be replaced by “Julio Cascante” but the same code [player] when inserted into a different file might get replaced with “Steve Clark.” This is a good time to point out the first basic when it comes to Hot Codes: The delimiter for Hot Codes are square brackets: [and]. (Unlike basic Code Replacements, these delimiters cannot be changed.)

History Break

You could accomplish some of this in the past in a roundabout way in Photo Mechanic by combining Variables with Code Replacements. A classic use of Code Replacements in the past involved editors working with a bucket of images from different photographers. Editors would combine Code Replacements with the {serial} variable in Photo Mechanic to sort images into different folders. All images that came from one camera would go to folder A, images that came from a second camera would go to folder B and so on. This was done by cleverly using the variable {serial} within Code Replacement delimiters. The edit might have a Code Replacement file with camera serial numbers as codes, and folders or names as replacements.

Example:

```
// Code Replacement file for Photo Mechanic
```

7654123	A	Walker
4325621	B	Orlosky
3212-1234	C	Keel

Within Photo Mechanic, the editor could select a large group of files, and in the Copy/Move dialog, tell PM to put the images into a folder with the string `={serial}=`. That's the serial number variable, inserted in between Code Replacement delimiters. For all images from the camera with serial number 7654123, the code `={serial}=` would be replaced with "A" and the code `={serial}#2=` would be replaced with "Walker" – useful for credit and keywords. But, for images from the camera with the serial number 4325621, the same codes `={serial}=` and `={serial}#2=` would be replaced with "B" and "Orlosky" respectively. This was very useful for many years and this sort of procedure is still very powerful when working with metadata. But, then along came one camera manufacturer who decided to put the camera serial number into the proprietary "maker note" section of their camera's image metadata, and this broke a lot of workflows for people who depended on that. That is where Hot Codes come in.

Hot Codes: Code Replacement Evolved

Hot Codes, then, can be described as a way to dynamically link Code Replacements with variables in the Code Replacement file itself, which enables a more powerful workflow, and when you understand how to use it, can actually make life easier because instead of remembering long strings like `={serial}#5=`, you can actually name your codes more clearly like `[folder]` and `[lastname]`. Traditional Code Replacements all still work as they always have, but in Photo Mechanic 6, Hot Codes were added and work in addition to the legacy Code Replacements.

General Advice

As we get into the actual demo and use of Hot Codes and Code Replacements, it is good to keep in mind that they do require some "upfront" work before you ingest the card, or ideally before you even go out shooting. The Code Replacement files and the Metadata Templates are all ideally prepped before the event. In the cases where the Hot Codes rely upon in-camera folders, you'll either want to create those folders in-camera before you even leave the house and/or keep your Code Replacement sheet handy when you are shooting so you'll know what folder to create or use. Keep in mind different manufacturers have different naming conventions for their folders, so

getting to know how your camera behaves will make this process run more smoothly for you. As with many things in life, doing a little extra work in preparation will reward you with much more time saved in the end.

Pro Tip: You can check or verify what the {folder} variable will be returning by navigating to the memory card in Photo Mechanic, opening up the Metadata Info panel for a photo on the card and putting {folder} into an unused field. Do NOT apply the Metadata Info, but hold down the Modifier key (Opt on macOS and Shift on Windows) and you will see the "OK" button change to "EVAL" this allows you to evaluate the variable or code replacement without actually applying it. Once you see what the variable is evaluating to, you should "Cancel" the Metadata Info without saving - thus preserving the images on your card.

The Hot Code Template

Reminder: This template sheet is available at this link:

https://docs.google.com/spreadsheets/d/17I0-VBZI-Qvdea8UxhyXozOEdejjZMmxv_YdIYdjQWU/edit?usp=sharing

//	Code Replacement file with Hot Codes for Photo Mechanic		
//	Enter the variables you will be looking for strings in		
//==	<i>variable1</i>		
//==	<i>variable2</i>		
//==	<i>variable3</i>		
//	Name your hotcodes here		
###	<i>hotcode1</i>	<i>hotcode2</i>	<i>hotcode3</i>
<i>string1</i>	<i>replacementA</i>	<i>replacementD</i>	<i>replacmentG</i>
<i>string2</i>	<i>replacementB</i>	<i>replacementE</i>	<i>replacementH</i>
<i>string3</i>	<i>replacementC</i>	<i>replacementF</i>	<i>replacementJ</i>
//??	<i>default</i>	<i>replacements</i>	<i>here</i>

The example above is a template for a Code Replacement file with Hot Codes, as it appears in the spreadsheet program, Google Sheets. There is some color coding to help illustrate different

sections, but when this sheet is exported as a plain-text .TSV file, it will not have any such formatting. Here is what the .tsv file looks like in a plain-text editor:

```

~/Downloads/Hot Codes Template - Sheet1.tsv
1 // Code Replacement file with Hot Codes for Photo Mechanic
2 // Enter the variables you will be looking for strings in
3 //== variable1
4 //== variable2
5 //== variable3
6 // Name your hotcodes here
7 //## hotcode1 hotcode2 hotcode3
8 string1 replacementA replacementD replacementG
9 string2 replacementB replacementE replacementH
10 string3 replacementC replacementF replacementJ
11 //?? default replacements here

```

This is where you need to polish your mental models to understand how Hot Codes work. If you're used to working with Code Replacements, you'll need to shift your thinking a little bit. In basic Code Replacements, the codes are in a vertical column on the left. In Hot Codes, the codes are all on a row prefixed by `//##` and what they get replaced by are in rows beneath them, controlled by the strings on the left.

Here are the basic notations you'll need to create Code Replacement files with Hot Codes

<code>//</code>	Any line that starts with // by itself will be ignored by Photo Mechanic so you can add comments into your file to help you remember what's going on.
<code>//==</code>	Followed by a TAB and Variable indicates what variables Photo Mechanic will look at to determine what to replace the Hot Code with. You may have a single line or many lines starting with this prefix with as many Variables as you need
<code>//##</code>	Denotes the row with all of your Hot Codes on it. This will generally appear just once, and your list of Hot Codes can extend out to the right of this (separated by TABs) for as long as you need. Rows beneath that will include the strings that Photo Mechanic will be looking for in the Variables specified above. For each string, which replacement to insert
<code>//??</code>	This row is used to specify what each Hot Code should get replaced with in case no strings matched any of the specified Variables. This "exception line" is used just once.
<code>//\$\$</code>	In cases when you want to load multiple Code Replacement files, each one will need to have its own "namespace" specified with this notation

Demo #1: Using Hot Codes to route files to specific folders

Here we have a folder of mixed photos from a team of many different photographers. We want to quickly sort them into different folders by photographer, and tag that photographer's name into the Credit field in the same step.

So we're creating the Hot Codes: [folder] and [name]

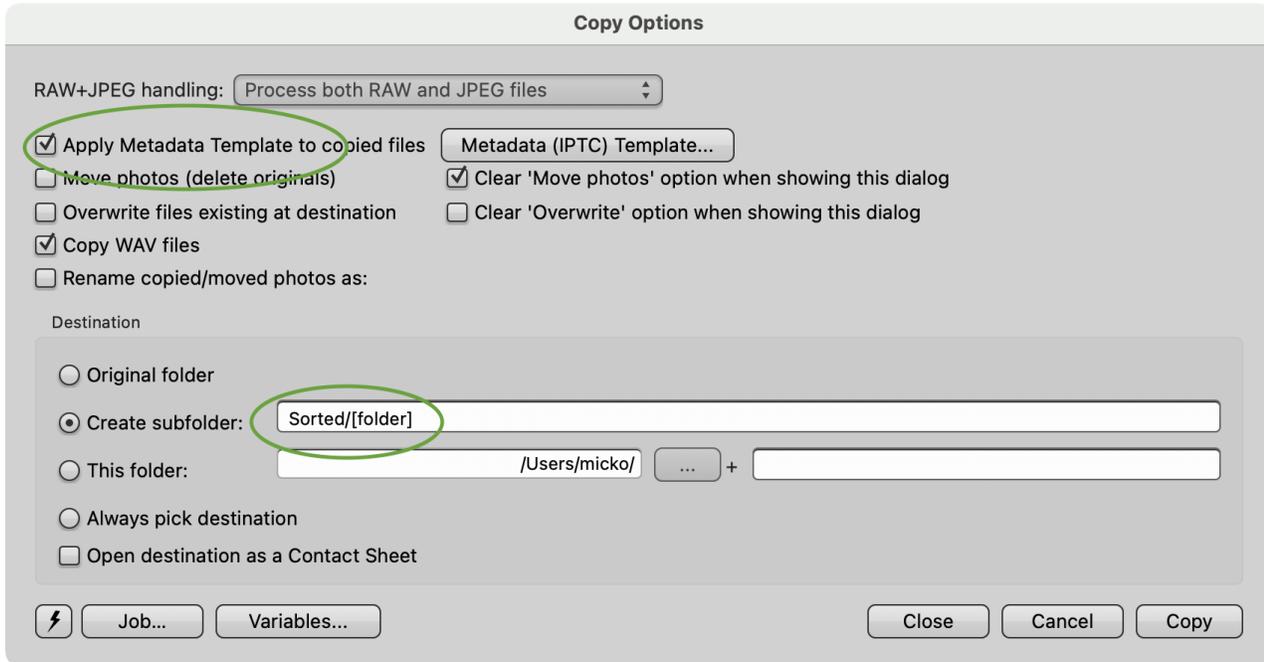
We know some of the serial numbers of the cameras used by the team, but for some we don't. However, for those whom we don't the serial number, we do know that Mick was using an iPhone 8, and Craig told us that his camera automatically writes his name into the Photographer field. So in our Code Replacement sheet, we'll specify which variables we're going to be looking at to determine what replacements get inserted for the codes [folder] and [name]. Those variables are {serial} {model} and {photog}

Code Replacement file with Hot Codes for Photo Mechanic			
Enter the variables you will be looking for strings in			
//==	{serial}		
//==	{model}		
//==	{photog}		
Name your hotcodes here			
//##	folder	name	
FFDT21819764	A	Mick O	
iPhone 8	A	Mick O	
03374267	B	Dennis W	
3008872	B	Dennis W	
Craig Mitcheldy	C	Craig M	
072024009290	D	Katy Z	
//??	E	Unknown	

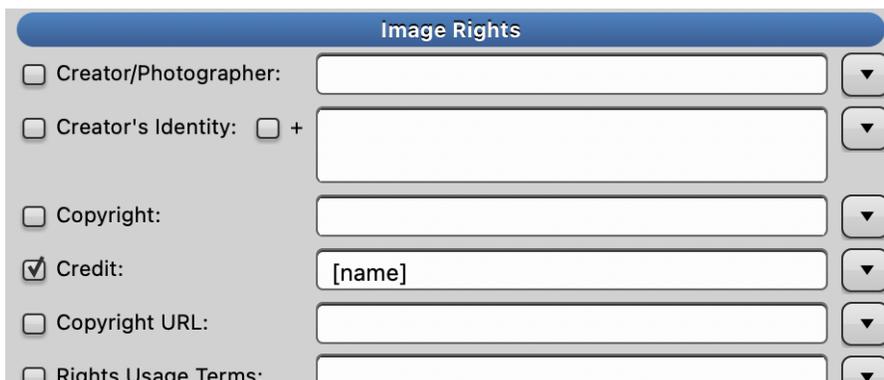
In the left column we'll place the information that we know we're looking for, and then what we'll want to Replace the Hot Codes with goes under those Hot Codes in their respective columns.

Once the sheet is complete, we'll save it as a plain-text .tsv (tab-separated value) file, then we'll load the Code Replacement file into Photo Mechanic under [Edit > Settings > Set Code Replacements](#).

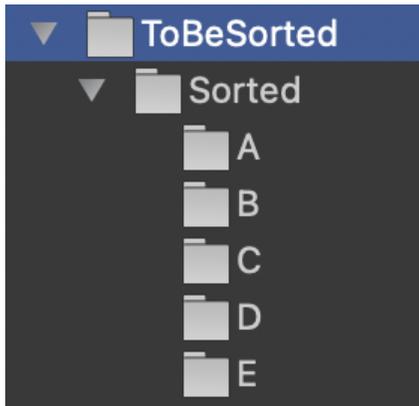
To sort the files, we'll open the **File > Copy/Move** dialog. When we're picking where to Copy or Move the photos, we can insert the Hot Code [folder] and know that Photo Mechanic will replace that with one of the values under "folder" in the Code Replacement sheet. In this case: A, B, C, D, or E.



We'll also apply a Metadata Template to the files as they are copied. In that Metadata Template, we're just going to put the Hot Code [name] in the Credit field. Photo Mechanic will replace [name] with one of the names under "name" in the Code Replacement Sheet.



Once we click the "Copy" button, the files will be copied into new folders in the "Sorted" subfolder.

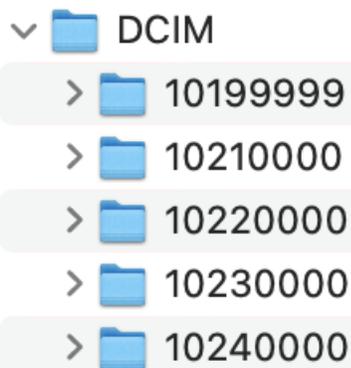


Demo #2: Applying metadata to files on ingest based on in-camera Folders

In this demo, we're going to be working with files on SD memory cards that have been recorded into folders on the card by using the camera's menu system. In Case 1, we'll look at sample files from a day shooting tennis. Before each match, the photographer has changed the destination folder in-camera so that each match is stored in its own folder. This early "pre-sort" of photos enables a great deal of automation that can be done in the Ingest process. Case 2 will involve a similar scenario where a photographer at a music festival has changed the in-camera folder before each set at various stages.

In both of these cases, we'll only need to specify one variable for Photo Mechanic to check to determine what values will be replaced. The variable `{folder}`.

In Case 1, the photographer has recorded each different match from a day at the tennis center to its own in-camera folder. Taking a look at the SD card in macOS Finder, we see that it looks like this:



For each different folder on the card, we can specify different replacements for the Hot Codes, [player1], [ctr1], [player2] et. al. Note that we can also specify basic Code Replacement codes in the same sheet.

Code Replacement file with Hot Codes for Photo Mechanic							
//	basic code replacements						
jobname	CitiOpen2022						
matchname	Citi Open Tennis Tournament						
matchvenue	Rock Creek Park Tennis Center						
matchcity	Washington, DC						
//	hot codes section						
//	Enter the variables you will be looking for strings in						
//==	{folder}						
//	Name your hot codes here						
//##	player1	ctr1	player2	ctr2	matchnum	key	cls
10210000	David Goffin	Belgium	Jack Sock	USA	first	goffinvsock	0
10220000	Alexei Popyrin	Australia	Taylor Fritz	USA	second	popyrinvfritz	0
10230000	Frances Tiafoe	USA	Botic Van De Zaandschulp	Netherlands	third	tiafoevvdz	0
10240000	Nick Kyrgios	Australia	David Goffin	Belgium	final	kyrgiosvgoffin	0
//??	a tennis player	a very nice n	another tennis player	a great country	this weekend's	needsreview	1

We'll save this Code Replacement file as a .tsv and eventually load it into Photo Mechanic. One thing to keep in mind. If you are crafting Metadata Templates with Code Replacements, it is sometimes best to leave the Code Replacement file unloaded until you have the Metadata Template set up the way you want it. You can use a mix of Variables, Code Replacements, and Hot Codes to craft this Metadata Template or something like it:

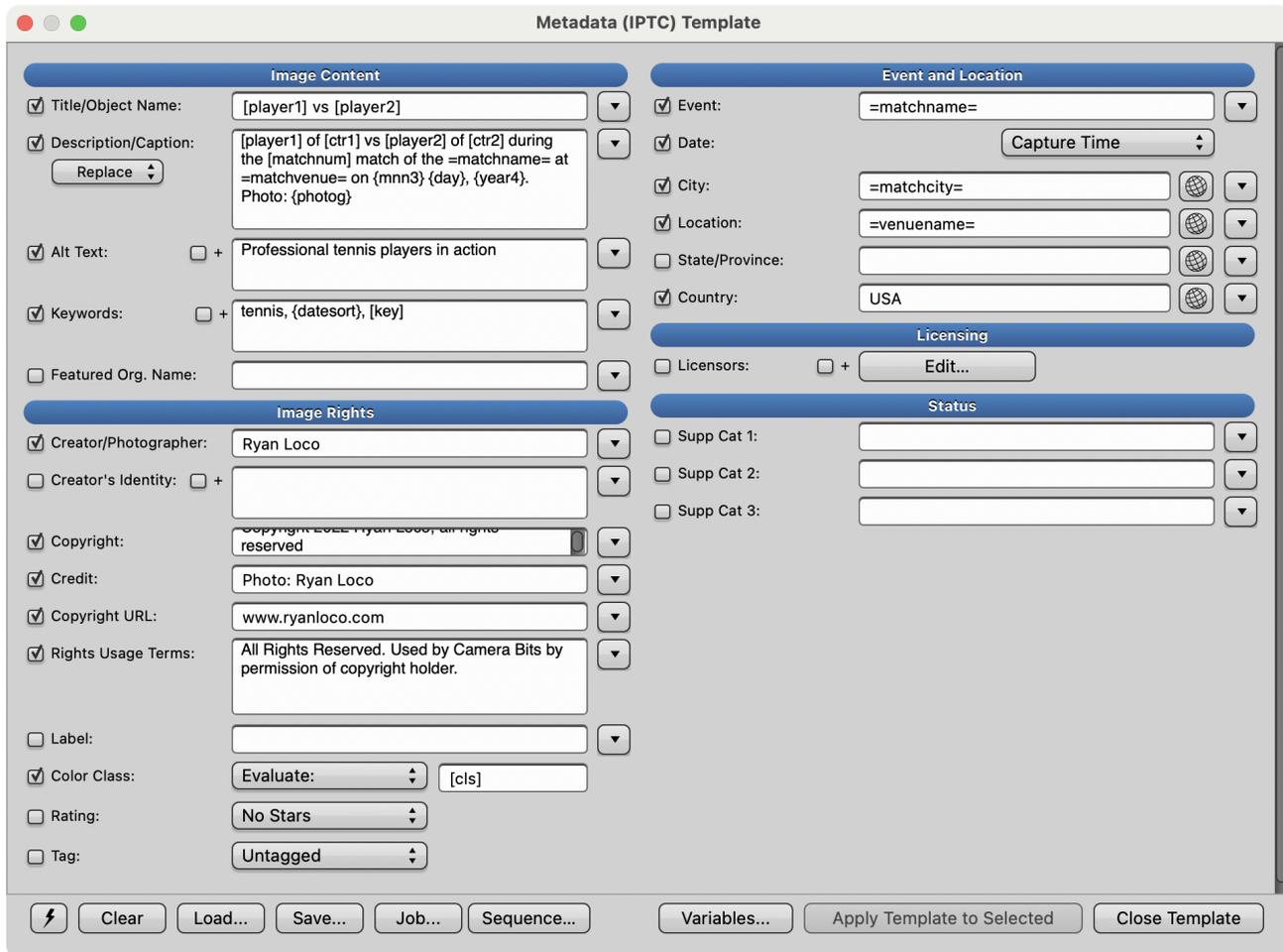


Image Content

- Title/Object Name: [player1] vs [player2]
- Description/Caption: [player1] of [ctr1] vs [player2] of [ctr2] during the [matchnum] match of the =matchname= at =matchvenue= on {mnn3} {day}, {year4}. Photo: {photog}
- Alt Text: Professional tennis players in action
- Keywords: tennis, {datesort}, [key]
- Featured Org. Name:

Image Rights

- Creator/Photographer: Ryan Loco
- Creator's Identity:
- Copyright: Copyright 2022 Ryan Loco, all rights reserved
- Credit: Photo: Ryan Loco
- Copyright URL: www.ryanloco.com
- Rights Usage Terms: All Rights Reserved. Used by Camera Bits by permission of copyright holder.
- Label:
- Color Class: Evaluate: [cls]
- Rating: No Stars
- Tag: Untagged

Event and Location

- Event: =matchname=
- Date: Capture Time
- City: =matchcity=
- Location: =venue=
- State/Province:
- Country: USA

Licensing

- Licensors: Edit...

Status

- Supp Cat 1:
- Supp Cat 2:
- Supp Cat 3:

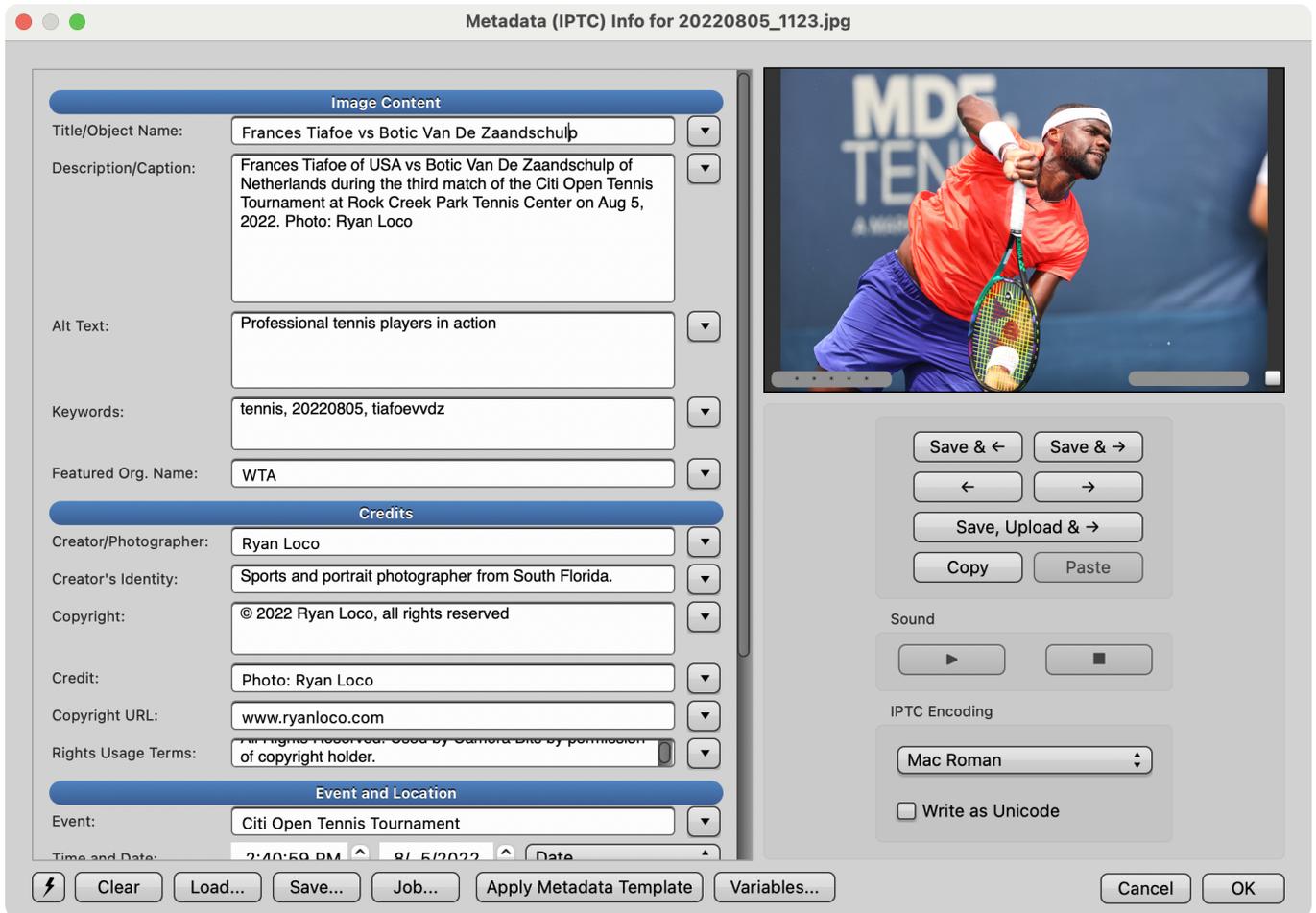
Buttons: Clear, Load..., Save..., Job..., Sequence..., Variables..., Apply Template to Selected, Close Template

Note the Color Class has been applied as "Evaluate:" with the Hot Code [cls]. From the Code Replacement sheet, we can see that [cls] will be replaced with 0 in all cases except the "exception" line for when no matches were found in the specified variables. The result of this will be that any files that were NOT in folders that we had accounted for will be given the Color Class of 1 (i.e. Red) and also will have the keyword "needsreview" so so that we can go back and clean up those files if needed.

Once you have set up and saved your Metadata Template, go ahead and load the Code Replacement file via Edit > Settings > Set Code Replacements. (Or get familiar with the keyboard shortcut: macOS: ⌘-Ctrl-c or Windows: Ctrl-Alt-c)

Eagle-eyed viewers will notice there was a folder on the SD card that we did not account for in the Code Replacement sheet. Maybe the photographer forgot to format the card, or perhaps there was an extra match added at the last minute. Either way, those files will appear with a Red Color Class

for special attention. Here is what the metadata for the files will look like after the Ingest process:



For Case 2, we will do something very similar for images from a music festival but add even more automation by sending images to folders based on the artist in the photo.

Again, here is the folder structure on the memory card. The photographer changed the recording folder in-camera before each set began.

- ▼  DCIM
 - >  101EOS5D
 - >  102EOS5D
 - >  103EOS5D
 - >  104EOS5D
 - >  105EOS5D

Here is the Code Replacement file for the music festival:

Code Replacement file with Hot Codes for Photo Mechanic				
// Basic code replacements				
eventname	<i>Pickathon</i>			
evcity	<i>Happy Valley</i>			
evstate	<i>OR</i>			
evcountry	<i>USA</i>			
// Enter the variables you will be looking for strings in				
//==	<i>{folder}</i>			
// Name your hot codes here				
//##	artist	stage	key	cls
102EOS5D	<i>Old Light</i>	<i>Galaxy Barn</i>	<i>oldlight</i>	<i>0</i>
103EOS5D	<i>Pure Bathing Cu</i>	<i>Galaxy Barn</i>	<i>purebathingculture</i>	<i>0</i>
104EOS5D	<i>Shakey Graves</i>	<i>Wood Barn</i>	<i>shakeygraves</i>	<i>0</i>
105EOS5D	<i>Andrew Bird</i>	<i>Mountain View Stage</i>	<i>andrewbird</i>	<i>3</i>
//??	scenes	grounds	needsreview	1

Again, we'll apply the red color class to files in unrecognized folders. This time we're also adding a green color class to images of the headliner to be able to find them more quickly. This is the Ingest dialog. Notice that this time, we're using a Hot Code [key] in the destination folder as well, so that each artist gets their own folder. Your workflow may or may not require this level of file routing, but we wanted to show that it is possible. We've also added much more information to the filename, including the Hot Code [artist]:

Ingest : 99 Concert - Pickathon

Source Paths

Ingest: Disks Folders From Selection

Card001 (864.8 MB used)

Card002 (7.4 GB used)

Incremental Ingest: copy new photos only
 Auto Catalog
 Auto Ingest

Source Directory Structure:

Copy Photos:

Folder Name:

Use folder sequence:

Primary Destination Path: /Users/micko/Pictures/Demo/HotCodesWebinar/{todaysort}/{job}/{key}

Maximum amount to transfer: 864.8 MB

Destination Folder Roots

Primary:

/Users/micko/Pictures/Demo/HotCodesWebinar/

Secondary:

Filter Files:

Apply Metadata Template to photos

Use Local Template
 Use Global Template

Rename Ingested photos as:

Sequence = 1136

Open Contact Sheets in background
 Erase Source Disk(s) after Ingest
 Unmount Source Disk(s) after Ingest

And here is the accompanying Metadata Template that will be applied on ingest:

Metadata (IPTC) Template

Image Content

Title/Object Name: [artist] at {event} on {mnn3} {day}, {year4}

Description/Caption: [artist performs at the [stage] at {event} {year4} in {city}, {state} on {mnn3} {day}.
Mandatory photo credit: {credit}

Alt Text: + Live music performers on stage

Keywords: + [key], {photog}, {model}, =sr{mn0}{day0}=, {datesort}

Featured Org. Name:

Event and Location

Event: =eventname=

Date: Capture Time

City: =evcity=

Location: [stage]

State/Province: =evstate=

Country: =evcountry=

Image Rights

Creator/Photographer: Mick Orlosky

Creator's Identity: + he/him, photographer of events in the Pacific Northwest

Copyright: © {year4} Mick Orlosky

Credit: Mick Orlosky / @redfishingboat

Copyright URL: https://redfishingboat.photoshelter.com/p/lice

Rights Usage Terms: Attribution-NonCommercial 4.0 International (CC BY-NC 4.0)
You are free to:

Label:

Color Class: Evaluate: [cls]

Rating: No Stars

Tag: Untagged

Licensing

Licensors: + Edit...

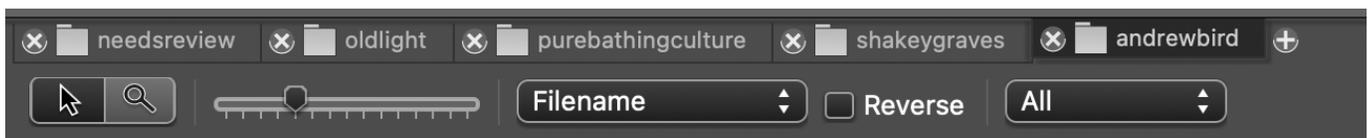
Status

Supp Cat 1:

Supp Cat 2:

Supp Cat 3:

The Ingest process puts images from each artist into their own folder and opens that folder in a tab in the Contact Sheet window.



Demo #3: Applying Metadata to Photos from a Spreadsheet

Another use case for Hot Codes is simplifying the process of taking a spreadsheet containing image metadata and applying that metadata to a group of images. Code Replacement with Hot Codes in Photo Mechanic can make this process very easy.

In this situation, you start with a folder of images with no metadata, but you have a spreadsheet that contains metadata associated with filenames. Such a spreadsheet may look something like this:

filename	title	caption	alttext	keywords
001_20191105.jpg	One Foggy Morning	St Johns Bridge in the fog	Black and white image of a road	sjb, fog, bw
002_20200131.jpg	SJB Fractured	St Johns Bridge through diffraction toy	Gothic spires seen through a toy	sjb, diffraction, toy
003_20200315.raf	Sunset Web	Sunset at Ridgefield Wildlife Park	Monochrome image of a sunset	ridgefield, sunset, silhouette
004_20210228.jpg	Escape To The Palms	Leave your cares behind and stay at The Palms.	A large yellow, brown, and green	palms, motel, hotel, neon, i
005_20210306.jpg	Polka Dots Downtown: Acoustic Minds	Portland's own Acoustic Minds played live in Pion	Wide angle image of a city skylin	pioneersquare, polkadots,
006_20210319.jpg	Plum Crazy	Start wearing purple	Close-up of the grille of an old pi	X-E4, plum, pickup, rust
007_20210322.jpg	Put On a Happy Face	It's not always the tears that define our pain, some	Close-up of a No Trespassing s	notrespassing, graffiti, spre
008_20210323.jpg	Chocolate Thunder	I am cautiously optimistic that I am starting to see	The tail end of a rusted old car w	thunderbird, basketball, ho
009_20210330.jpg	Never Felt Further Away	A stream of consciousness in a vanishing industria	A railyard behind a cyclone fence	train, fence, swanislund, ve

Well, with a list of filenames on the left, and descriptive column headers in a row, you've nearly got a workable Code Replacement file. You only need to add a few things like `//==` to tell what variable to look at, in this case `{filename}` and then `//##` to define the column headers as your Hot Codes

<code>//==</code>	<code>{filename}</code>			
<code>//##</code>	title	caption	alttext	keywords
001_20191105.jpg	One Foggy Morning	St Johns Bridge in the fog	Black and white	sjb, fog, b
002_20200131.jpg	SJB Fractured	St Johns Bridge through diffraction toy	Gothic spires se	sjb, diffrac
003_20200315.raf	Sunset Web	Sunset at Ridgefield Wildlife Park	Monochrome im	ridgefield,
004_20210228.jpg	Escape To The Palms	Leave your cares behind and stay at The Palms.	A large yellow, b	palms, mc
005_20210306.jpg	Polka Dots Downtown: Acoustic Minds in Pi	Portland's own Acoustic Minds played live in Pioneer Squa	Wide angle imag	pioneersq
006_20210319.jpg	Plum Crazy	Start wearing purple	Close-up of the	X-E4, plur
007_20210322.jpg	Put On a Happy Face	It's not always the tears that define our pain, sometimes it	Close-up of a Nc	notrespas

As you can see, the "exception row" is not actually required. You can add one if you want, but we're going to demo this one as "quick n' dirty". Preparing a Metadata Template to apply this to your photos is incredibly simple:

Metadata (IPTC) Template

Image Content

Title/Object Name: ▼

Description/Caption: ▼
 ▼

Alt Text: + ▼

Keywords: + ▼

Featured Org. Name: ▼

Image Rights

Creator/Photographer: ▼

Creator's Identity: + ▼

Copyright: ▼

Now save your spreadsheet as a plain-text .tsv and load it into Photo Mechanic. Remember: [Edit > Settings > Set Code Replacements](#). Then apply the Metadata Template and see the results:

Metadata (IPTC) Info for 012_20210502.jpg

Image Content

Title/Object Name:

Description/Caption:

Alt Text:

Keywords:

Featured Org. Name:

Credits

Creator/Photographer:

Creator's Identity:

Copyright:

Credit:

Copyright URL:

Rights Usage Terms:

Event and Location

Event:

Time and Date:



Save & ← Save & →

← →

Save, Upload & →

Copy Paste

Sound

▶ ■

IPTC Encoding

Mac Roman

Write as Unicode

⚡ Clear Load... Save... Job... Apply Metadata Template Variables... Cancel OK